REMARKS

Claims 1 and 3-13 remain in the application and have been finally rejected. Claims 1, 12, and 13 have been amended. Claim 2 has been cancelled. Applicant respectfully requests reconsideration. This amendment should be entered because it is an amendment presenting rejected claims in better form for consideration on appeal. See 37 CFR 1.116(b)(2).

CLAIM REJECTIONS UNDER 35 USC §102

The Office Action rejected claims 1 and 3-13 under 35 USC 102 as anticipated by U.S. Patent No. 6,061,679 issued to Bournas (hereafter, "Bournas"). Anticipation requires that each and every element of the claimed invention be disclosed in a single reference of prior art. *In re Paulsen*, 30 F.3d 1475 (Fed. Cir. 1994). The claim elements must be arranged as in the claim. *Richardson v. Suzuki Motor Co.*, 868 F.2d 1226 (Fed. Cir. 1989). For anticipation there must be **no difference** between the claimed invention and the reference disclosure. *Scripps Clinic & Res. Found. V. Genetech, Inc.*, 927 F.2d 1565 (Fed. Cir. 1991).

Claim 1 is not anticipated by Bournas because Bournas does not disclose the claimed step of "periodically traversing selected subgraphs of the region in the application <u>in order to detect data structure changes in the subgraphs</u> while the application is running." Bournas relates to a method for creating, modifying and searching a data structure and not to a method of *determining* how a region of a data structure in an application evolves. The cited

parts of Bournas [col. 4, lines 45-47, 61-65, col. 7, line 66 – col. 8, lines 39-66] do not discuss detecting changes in a data structure, let alone while the application is running.

The cited part of Bournas reads as follows:

"The data structure of the present invention, as well as the inventive techniques for creating, modifying and searching the data structure, are applicable to many situations including, but not limited to, routing, in general; the routing of requests according to network addresses used by the Internet Protocol (IP) (The IP protocol is used by Transmission Control Protocol (TCP)/IP set of protocols and the Open Systems Internet (OSI) TP set of protocols.); routing of logical units (LUs) used by the IBM Systems Network Architecture (SNA) protocol; and the obtaining of information for users of, for instance, Telenet." Col. 4, lines 45-47.

"In accordance with the principles of the present invention, a data structure is created that includes a plurality of ordered sub-data structures. One example of a data structure 200 is depicted in FIG. 2. As shown, data structure 200 includes a plurality of sub-data structures 202." Col. 4, lines 61-65

"In accordance with the principles of the present invention, the data structure (e.g., the routing table) can be dynamically modified by adding keys (e.g., addresses) of addressable elements or subnets to the data structure. One embodiment for inserting into a data structure is described below with reference to FIGS. 4a-4b.

"Initially, an add request is received by a computing unit that will use the request to update the data structure, STEP 400." Col. 7, line 66 – col. 8, line 6.

Subsequent to calculating the key mask range, the data structure is searched to determine if there is a tree root node for this key mask range, STEP 408. Specifically, the key mask range stored within each root node is checked to see if one matches the key mask range just calculated. If the there is no root node for this particular key mask range, INQUIRY 410, then a new root node is created, STEP 412. Thereafter, the new root node is inserted into a linked list of trees (which make up the data structure) according to its range, STEP 414. In particular, the smaller the key mask range, the more specific the subnet is, and the more specific the key mask is, the closer to the beginning of the data structure the subnet is placed. Thus, the data structure starts with the most specific key mask range, range one, and proceeds, for instance, in ascending order of key mask ranges.

After inserting the new root node into the data structure according to its range, the new key mask is stored in the root node, STEP 416.

Returning to INQUIRY 410, if there is a root for this key mask range, then a further determination is made as to whether this particular key mask is among the stored key masks of the found root node, INQUIRY 418. Should the key mask not be included in the stored key masks, then it is added, STEP 420. After the key mask is added to the root node (STEPS 416, 420) or if the key mask is already among the stored key masks (INQUIRY 418), then the key (e.g., the address) is added to the found tree, STEP 422." Col. 8, lines 39-66.

This says nothing about determining changes in a substructure or about doing that while the application is running and there is no explanation in the Office Action on how the Examiner considers that the required teaching occurs.

Bournas does not disclose "locating structural changes in the subgraphs" as claimed. The Office Action cites the following for this limitation:

"Subsequent to calculating the key mask range, the data structure is searched to determine if there is a tree root node for this key mask range, STEP 408. Specifically, the key mask range stored within each root node is checked to see if one matches the key mask range just calculated. If the there is no root node for this particular key mask range, INQUIRY 410, then a new root node is created, STEP 412. Thereafter, the new root node is inserted into a linked list of trees (which make up the data structure) according to its range, STEP 414. In particular, the smaller the key mask range, the more specific the subnet is, and the more specific the key mask is, the closer to the beginning of the data structure the subnet is placed. Thus, the data structure starts with the most specific key mask range, range one, and proceeds, for instance, in ascending order of key mask ranges.

After inserting the new root node into the data structure according to its range, the new key mask is stored in the root node, STEP 416.

Returning to INQUIRY 410, if there is a root for this key mask range, then a further determination is made as to whether this particular key mask is among the stored key masks of the found root node, INQUIRY 418. Should the key mask not be included in the stored key masks, then it is added, STEP 420. After the key mask is added to the root node (STEPS 416, 420) or if the key mask is already among the stored key masks (INQUIRY 418), then the key (e.g., the address) is added to the found tree, STEP 422."

There is no discussion of structural changes in the cited text above.

According to Bournas a sub-data structure 202 is denoted as 22 in Fig. 2. Item 22 is shown as a node. A node is not a sub-graph. See specification at page 9, lines 20-22, where a subgraph is defined to include nodes and remaining constituents. A graph comprises nodes and edges. The definition of a subgraph is:

Sub-Graph. A sub-graph is a subset of a graph *G* where *p* is the number of sub-graphs. For instance *G' = (v', e')* can be a distinct sub-graph of *G*. Unless the global transport system is considered in its whole, every transport network is in theory a sub-graph of another. For instance, the road transportation network of a city is a sub-graph of a regional transportation network, which is itself a sub-graph of a national transportation network.

See Graph Theory: Definition and Properties, Jean-Paul Rodrigue (2006). Traversing a node is a meaningless statement. Therefore, a subgraph is not a sub-data structure.

Bournas does not disclose "using these structural changes to describe, characterize, and identify changes to the region as a whole." The Office Action contends that adding keys of addressable elements to the data structure corresponds to this limitation. However, Bournas does not do this for a region of a data structure.

Finally, Bournas does not disclose reporting the changes to the region to an analysis agent. Bournas at col. 9, lines 12-13 says that the requester of the addition receives a report of a *duplication error*. That is not the same as reporting changes to the region to an analysis agent.

Claims 3-11 are dependent on claim1 and are not anticipated for the above

reasons. Claims 12 and 13 are, respectively, program product manufacture and machine counterparts of claim 1 and are also not anticipated for the above reasons.

For the foregoing reasons, Applicant respectfully allowance of the pending claims.

Respectfully submitted,

Michael J. Buchenhorner

Michael J. Buchenhorner
Reg. No. 33,162

E-filed on Date: June 25, 2007

Michael Buchenhorner, P.A.
8540 S.W. 83 Street
Miami, Florida 33143
(305) 273-8007 (voice)
(305) 595-9579 (fax)